```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv('/content/compressed_data.csv')
```

```
<ipython-input-3-88a8be71c876>:1: DtypeWarning: Columns (25) have
mixed types. Specify dtype option on import or set low_memory=False.
  df=pd.read_csv('/content/compressed_data.csv')
```

```
df.head(5)
```

```
{"type":"dataframe","variable_name":"df"}
```

questions to solve:

1. what is the distribution of listing prices?
2. how are the different room types distributed?
3. how are listings distributed across different neighborhoods?
4. what is the relationship between price and room type?
5. how has the number of reviews changed over time?

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 26 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102599 non-null  int64
 1   NAME                            102349 non-null  object
 2   host id                         102599 non-null  int64
 3   host_identity_verified          102310 non-null  object
 4   host name                       102193 non-null  object
 5   neighbourhood group             102570 non-null  object
 6   neighbourhood                   102583 non-null  object
 7   lat                             102591 non-null  float64
 8   long                            102591 non-null  float64
 9   country                         102067 non-null  object
 10  country code                    102468 non-null  object
 11  instant_bookable                102494 non-null  object
 12  cancellation_policy             102523 non-null  object
 13  room type                       102599 non-null  object
 14  Construction year               102385 non-null  float64
 15  price                           102352 non-null  object
 16  service fee                     102326 non-null  object
 17  minimum nights                  102190 non-null  float64
 18  number of reviews               102416 non-null  float64
 19  last review                     86706 non-null   object
```

```
 20  reviews per month               86720 non-null   float64
 21  review rate number              102273 non-null  float64
 22  calculated host listings count  102280 non-null  float64
 23  availability 365                102151 non-null  float64
 24  house_rules                     50468 non-null   object
 25  license                         2 non-null       object
dtypes: float64(9), int64(2), object(15)
memory usage: 20.4+ MB
```

```
df.isnull().sum()
```

```
id                                    0
NAME                                250
host id                               0
host_identity_verified              289
host name                           406
neighbourhood group                  29
neighbourhood                        16
lat                                   8
long                                  8
country                             532
country code                        131
instant_bookable                    105
cancellation_policy                  76
room type                             0
Construction year                   214
price                               247
service fee                         273
minimum nights                      409
number of reviews                   183
last review                       15893
reviews per month                 15879
review rate number                  326
calculated host listings count      319
availability 365                    448
house_rules                       52131
license                          102597
dtype: int64
```

Handling missing values

```
df['last review']=pd.to_datetime(df['last review'],errors='coerce')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 26 columns):
```

```
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102599 non-null  int64
 1   NAME                            102349 non-null  object
 2   host id                         102599 non-null  int64
 3   host_identity_verified          102310 non-null  object
 4   host name                       102193 non-null  object
 5   neighbourhood group             102570 non-null  object
 6   neighbourhood                   102583 non-null  object
 7   lat                             102591 non-null  float64
 8   long                            102591 non-null  float64
 9   country                         102067 non-null  object
 10  country code                    102468 non-null  object
 11  instant_bookable                102494 non-null  object
 12  cancellation_policy             102523 non-null  object
 13  room type                       102599 non-null  object
 14  Construction year               102385 non-null  float64
 15  price                           102352 non-null  object
 16  service fee                     102326 non-null  object
 17  minimum nights                  102190 non-null  float64
 18  number of reviews               102416 non-null  float64
 19  last review                     86706 non-null   datetime64[ns]
 20  reviews per month               86720 non-null   float64
 21  review rate number              102273 non-null  float64
 22  calculated host listings count  102280 non-null  float64
 23  availability 365                102151 non-null  float64
 24  house_rules                     50468 non-null   object
 25  license                         2 non-null       object
dtypes: datetime64[ns](1), float64(9), int64(2), object(14)
memory usage: 20.4+ MB
```

Filling null values with zeroes and filling last review null values with minimum values.

```
df.fillna({'reviews per month':0,'last review':df['last
review'].min()},inplace=True)

df.dropna(subset=['NAME','host name'],inplace=True)

print(df.isnull().sum())
```

```
id                                0
NAME                              0
host id                           0
host_identity_verified          276
host name                         0
neighbourhood group              26
neighbourhood                    16
lat                               8
long                              8
country                         526
```

```
country code                            122
instant_bookable                         96
cancellation_policy                      70
room type                                 0
Construction year                       200
price                                   239
service fee                             268
minimum nights                          403
number of reviews                       182
last review                               0
reviews per month                         0
review rate number                      314
calculated host listings count          318
availability 365                        420
house_rules                           51867
license                              101947
dtype: int64
```

```
df.drop(columns=["house_rules","license"],inplace=True)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 101949 entries, 0 to 102598
Data columns (total 24 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              101949 non-null  int64
 1   NAME                            101949 non-null  object
 2   host id                         101949 non-null  int64
 3   host_identity_verified          101673 non-null  object
 4   host name                       101949 non-null  object
 5   neighbourhood group             101923 non-null  object
 6   neighbourhood                   101933 non-null  object
 7   lat                             101941 non-null  float64
 8   long                            101941 non-null  float64
 9   country                         101423 non-null  object
 10  country code                    101827 non-null  object
 11  instant_bookable                101853 non-null  object
 12  cancellation_policy             101879 non-null  object
 13  room type                       101949 non-null  object
 14  Construction year               101749 non-null  float64
 15  price                           101710 non-null  object
 16  service fee                     101681 non-null  object
 17  minimum nights                  101546 non-null  float64
 18  number of reviews               101767 non-null  float64
 19  last review                     101949 non-null  datetime64[ns]
 20  reviews per month               101949 non-null  float64
 21  review rate number              101635 non-null  float64
 22  calculated host listings count  101631 non-null  float64
```

```
 23  availability 365                101529 non-null  float64
dtypes: datetime64[ns](1), float64(9), int64(2), object(12)
memory usage: 19.4+ MB
```

changing data type of "price & service fee"

```
df['price']=df['price'].replace('[\$,]','',regex=True).astype(float)
df['service fee']=df['service fee'].replace('[\
$,]','',regex=True).astype(float)

df.head(5)
```


```
{"type":"dataframe","variable_name":"df"}
```


```
df.drop_duplicates(inplace=True)
```

Descriptive Statistics

```
df.describe()
```

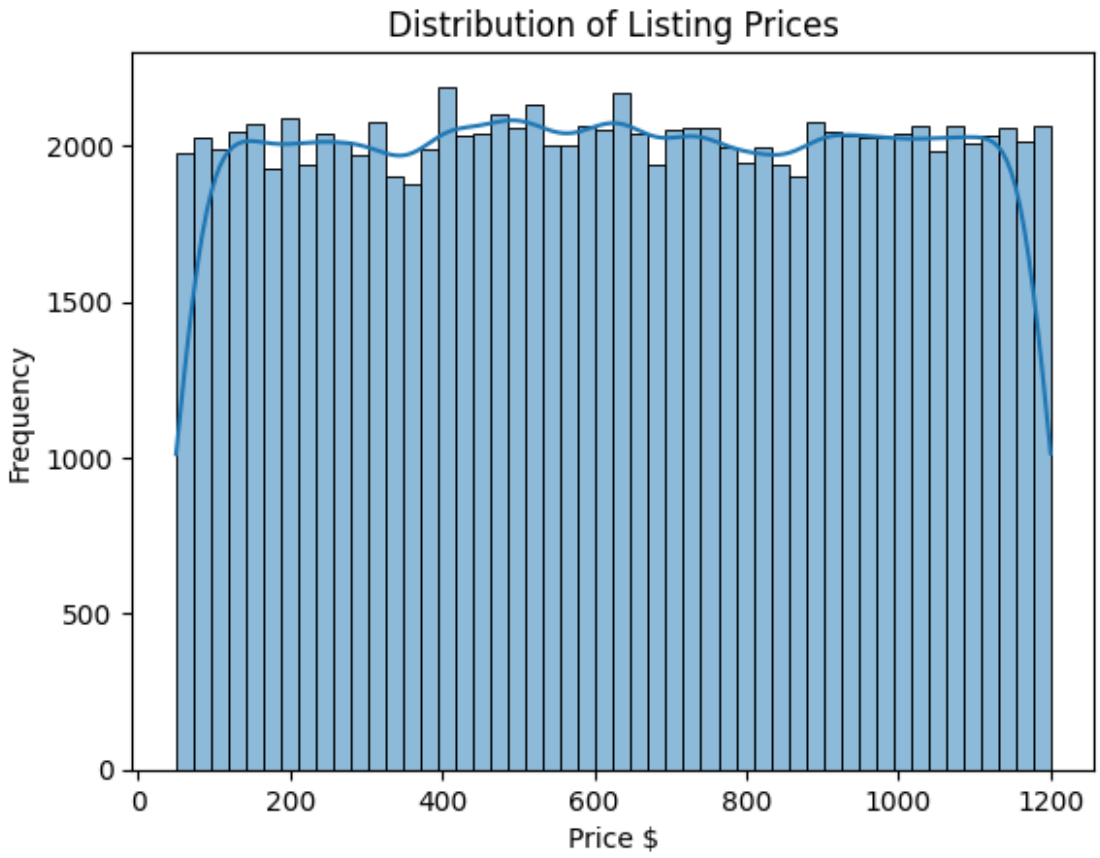

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 19925526.64955548,\n
\"min\": 101410.0,\n        \"max\": 57367417.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
29209587.716803078,\n          43283077.0,\n          101410.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"host id\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
34505511109.68557,\n        \"min\": 101410.0,\n        \"max\":
98763129024.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          49261546471.41352,\n
73997470050.25,\n          101410.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"lat\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 35838.680963229155,\n
\"min\": 0.055849709175834955,\n        \"max\": 101402.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
40.72808189059348,\n          40.76275,\n          101402.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"long\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 35873.441738603426,\n        \"min\": -74.24984,\n
\"max\": 101402.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          -73.94966263900959,\n          -73.93234,\n
101402.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Construction year\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 35180.05194379358,\n        \"min\":
5.765129625298182,\n        \"max\": 101210.0,\n
```

```
\"num_unique_values\": 8,\n        \"samples\": [\n
2012.48690840826,\n          2017.0,\n          101210.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"price\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 35564.83349159863,\n
\"min\": 50.0,\n        \"max\": 101171.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
625.3810083917328,\n          913.0,\n          101171.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"service fee\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
35717.88630766372,\n        \"min\": 10.0,\n        \"max\":
101142.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
125.04399754800183,\n          183.0,\n          101142.0\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"minimum nights\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
35550.10557292513,\n        \"min\": -1223.0,\n        \"max\":
101016.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
8.113744357329532,\n          5.0,\n          101016.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"number of reviews\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
35733.65418729364,\n        \"min\": 0.0,\n        \"max\": 101228.0,\
n        \"num_unique_values\": 8,\n        \"samples\": [\n
27.51185442762872,\n          31.0,\n          101228.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"last review\",\n
\"properties\": {\n        \"dtype\": \"date\",\n        \"min\":
\"1970-01-01 00:00:00.000101410\",\n        \"max\": \"2058-06-16
00:00:00\",\n        \"num_unique_values\": 7,\n        \"samples\":
[\n          \"101410\",\n          \"2018-05-15
21:26:08.721033728\",\n          \"2019-07-01 00:00:00\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"reviews per month\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
35849.0582833207,\n        \"min\": 0.0,\n        \"max\": 101410.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
1.1632067843407947,\n          1.71,\n          101410.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"review rate number\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
35744.320347080014,\n        \"min\": 1.0,\n        \"max\":
101103.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
3.2785575106574485,\n          4.0,\n          101103.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"calculated host listings count\",\
n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 35722.54594272106,\n        \"min\": 1.0,\n        \"max\":
```

```
101092.0,\n        \"num_unique_values\": 6,\n        \"samples\": [\n
101092.0,\n          7.948462786372809,\n          32.3289735638349\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"availability 365\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
35509.691706077516,\n        \"min\": -10.0,\n        \"max\":
100990.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
141.1646598673136,\n          269.0,\n          100990.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


1. what is the distribution of listing prices?

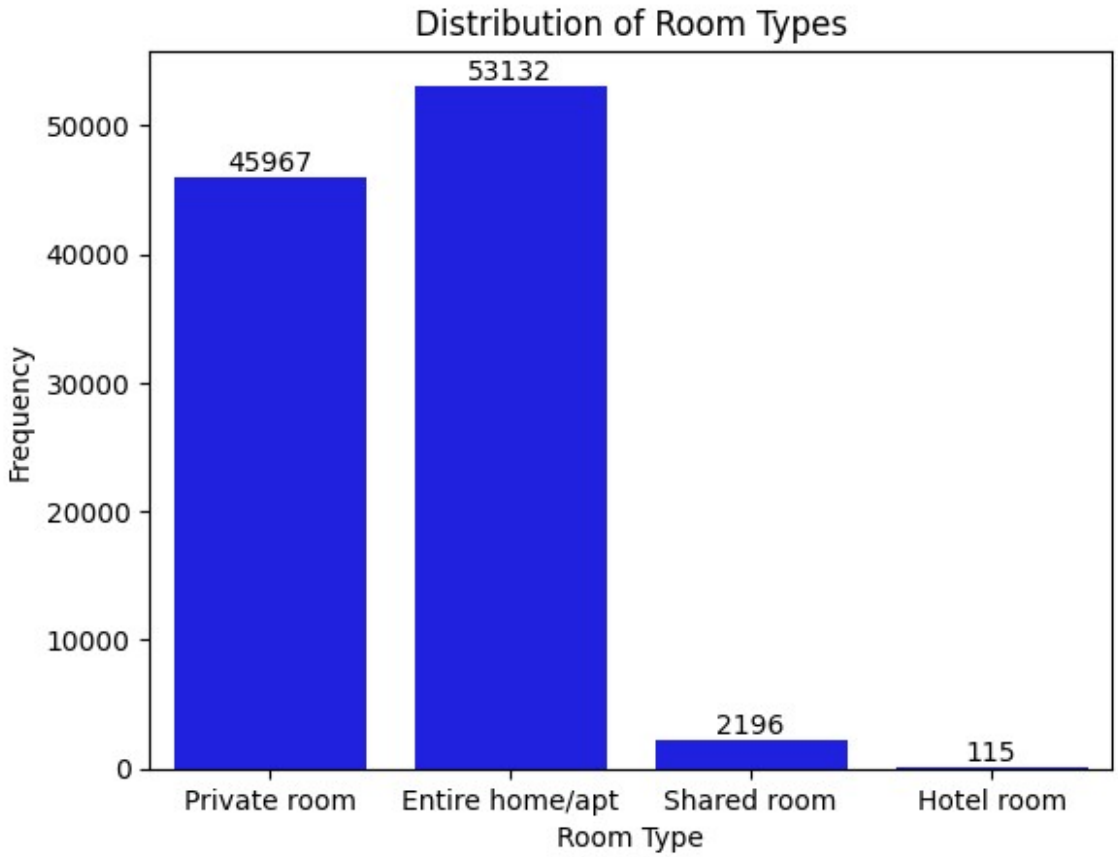
```
sns.histplot(df['price'],bins=50,kde=True)
plt.title('Distribution of Listing Prices')
plt.xlabel('Price $')
plt.ylabel('Frequency')
plt.show()
```



1. how are the different room types distributed?

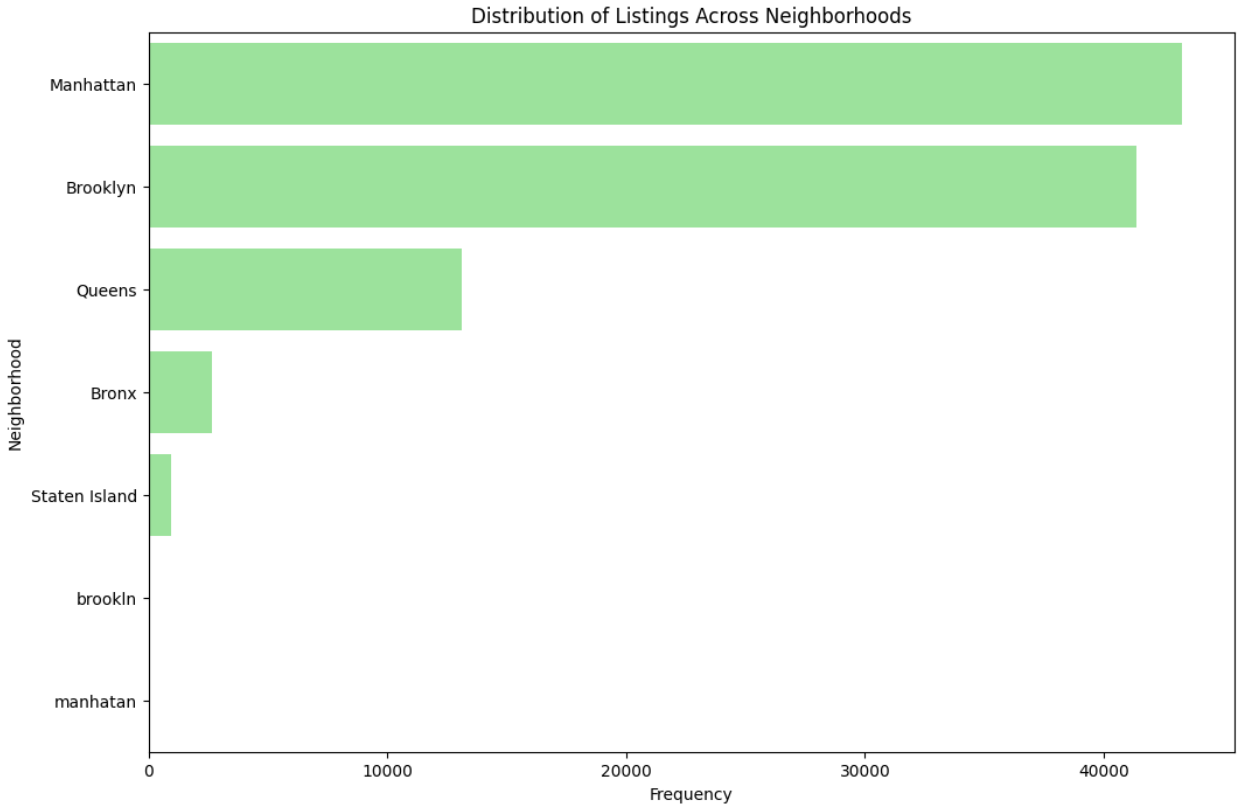
```
ax=sns.countplot(x='room type',data=df,color='blue')
ax.bar_label(ax.containers[0])
```

```
plt.title('Distribution of Room Types')
plt.xlabel('Room Type')
plt.ylabel('Frequency')
plt.show()
```



1. how are listings distributed across different neighborhoods?

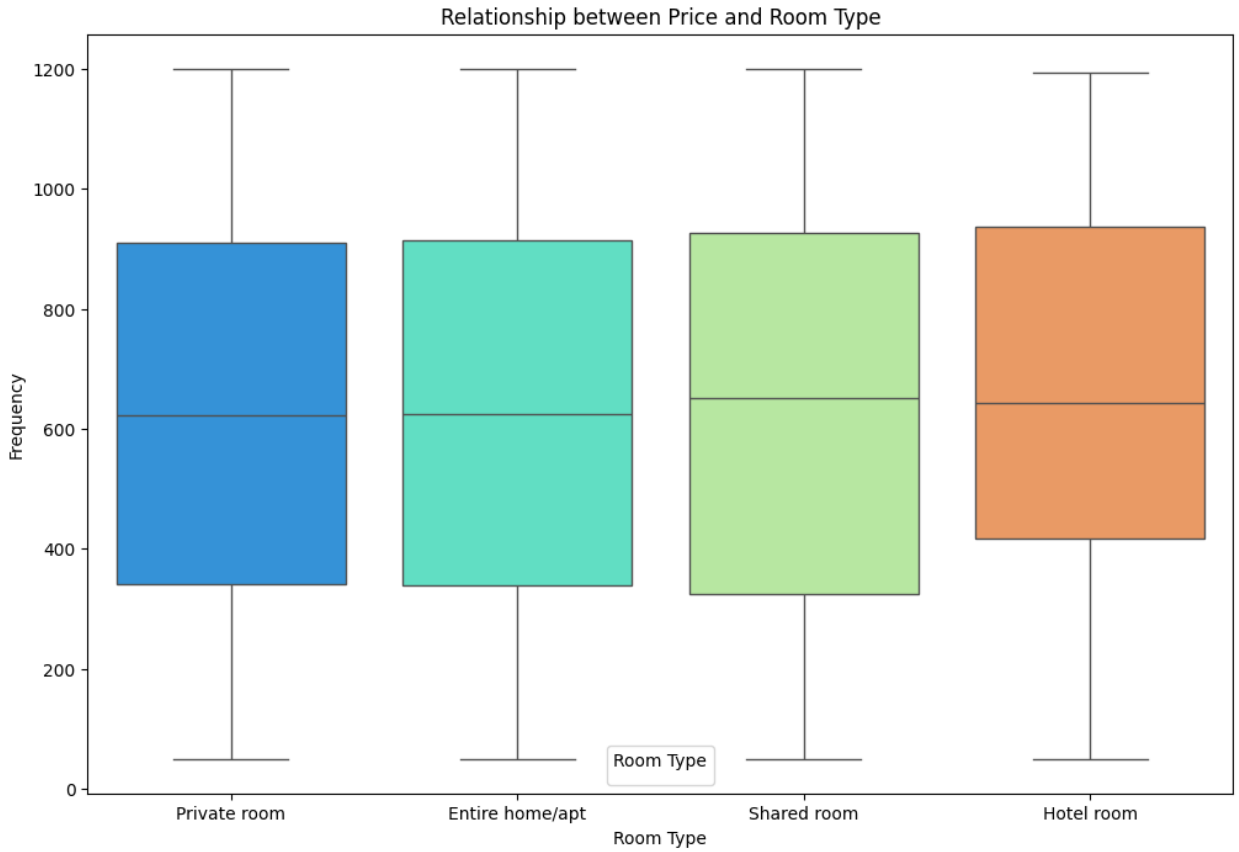
```
plt.figure(figsize=(12,8))
sns.countplot(y='neighbourhood
group',data=df,color='lightgreen',order=df['neighbourhood
group'].value_counts().index)
plt.title('Distribution of Listings Across Neighborhoods')
plt.xlabel('Frequency')
plt.ylabel('Neighborhood')

Text(0, 0.5, 'Neighborhood')
```



1. what is the relationship between price and room type?

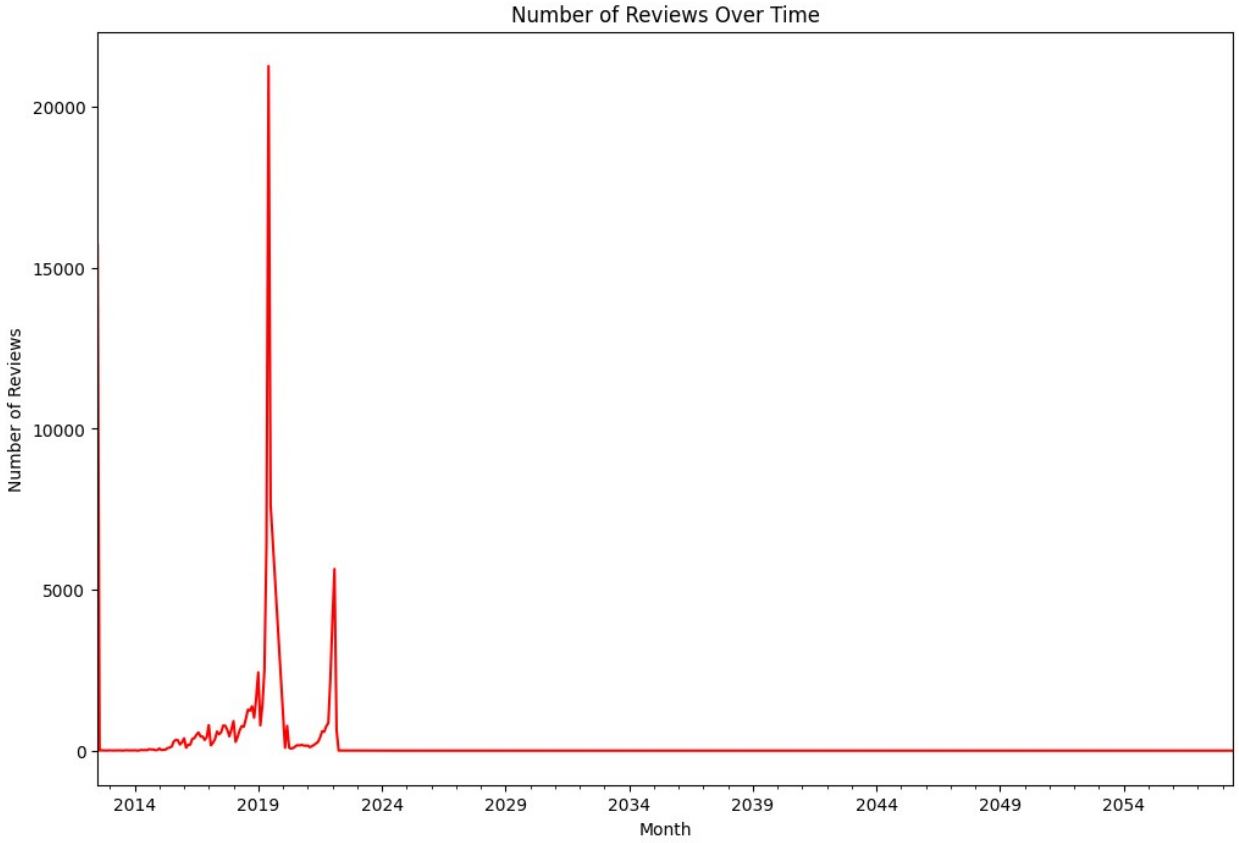
```
plt.figure(figsize=(12,8))
sns.boxplot(x='room type',y='price',hue='room
type',data=df,palette='rainbow')
plt.title('Relationship between Price and Room Type')
plt.xlabel('Room Type')
plt.ylabel('Frequency')
plt.legend(title='Room Type')
plt.show()
```

```
<ipython-input-55-3e74c957b928>:6: UserWarning: No artists with labels
found to put in legend.  Note that artists whose label start with an
underscore are ignored when legend() is called with no argument.
  plt.legend(title='Room Type')
```



1. how has the number of reviews changed over time?

```
df['last review']=pd.to_datetime(df['last review'])
reviews_over_time=df.groupby(df['last
review'].dt.to_period('M')).size()

plt.figure(figsize=(12,8))
reviews_over_time.plot(kind='line',color='red')
plt.title('Number of Reviews Over Time')
plt.xlabel('Month')
plt.ylabel('Number of Reviews')
plt.show()
```


Number of Reviews Over Time
Number of Reviews
20000
15000
10000
5000
0
2014
2019
2024
2029
2034
2039
2044
2049
2054
Month